

بسمه تعالی



گزارش کار اول آزمایشگاه مدارهای منطقی

آشنایی با محیط‌های شبیه‌سازی

دکتر حسابی

نویسنده

مریم شیران

دانشگاه صنعتی شریف

بهار ۱۴۰۱-۱۴۰۲

فهرست مطالب

۱	آزمایش اول: رسم مدار با Fritzing	۱
۱-۱	مقدمه و هدف	۱
۲-۱	تجزیه و تحلیل تئوری آزمایش	۱
۳-۱	شرح دستگاه‌ها و وسایل مورد استفاده	۱
۴-۱	روش انجام آزمایش	۲
۱-۴-۱	بخش اول	۲
۲-۴-۱	بخش دوم	۴
۳-۴-۱	بخش سوم	۵
۵-۱	بحث و تفسیر اطلاعات و نتایج به دست آمده از آزمایش	۶
۶-۱	منابع و مراجع	۸
۲	آزمایش دوم: ساخت مدار با Logisim	۸
۱-۲	مقدمه و هدف	۸
۲-۲	تجزیه و تحلیل تئوری آزمایش‌ها	۸
۳-۲	شرح دستگاه‌ها و وسایل مورد نیاز	۸
۴-۲	روش انجام آزمایش	۸
۵-۲	بحث و تفسیر اطلاعات نتایج به دست آمده از آزمایش	۱۱
۶-۲	منابع و مراجع	۱۲
۳	آزمایش سوم: ساخت مدار با Proteus	۱۳
۱-۳	مقدمه و هدف	۱۳
۲-۳	تجزیه و تحلیل تئوری آزمایش	۱۳
۳-۳	شرح دستگاه‌ها و وسایل مورد نیاز	۱۴
۴-۳	روش انجام آزمایش	۱۴

۳-۵ بحث و تفسیر نتایج به دست آمده از آزمایش.....۱۵

۳-۶ منابع و مراجع.....۱۶

۱ آزمایش اول: رسم مدار با Fritzing

۱-۱ مقدمه و هدف

در این قسمت به دنبال آنیم که چند مدار ساده را با نرم افزار Fritzing (یک نرم‌افزار شبیه سازی) طراحی کنیم.
هدف از آزمایش، آشنایی با چگونگی کار با نرم‌افزار Fritzing می باشد.

۱-۲ تجزیه و تحلیل تئوری آزمایش

در قسمت اول، با کمک نرم افزار اتصالات برد برد را بررسی میکنیم. برد برد ابزاری بسیار ساده و مناسب برای پیاده سازی مدارهای آزمایشی است که نیاز به هیچ گونه لحیم کاری ندارد. این وسیله در حقیقت یک پیش نمونه ساز است یعنی به کمک آن می توانیم یک نمونه ابتدایی از آن چه که قرار است در نهایت ساخته شود ایجاد کنیم. انتظار میرود که قسمت‌های بالا و پایین بردبرد به صورت افقی و قسمت‌های میانی بردبرد به صورت عمودی به یکدیگر متصل باشند.

در قسمت دوم، یک مدار ساده شامل باتری، مقاومت و LED را با روی بردبرد طراحی کنیم، از آنجا که نرم‌افزار Fritzing قابلیت شبیه سازی ندارد نمیتوانیم از صحت کار مدار مطلع شویم ولی اتصالات را بررسی میکنیم تا به روش دلخواه ما باشند.

در قسمت سوم، با استفاده از یک تراشه که شامل ۶ گیت نات میباشد (مانند تراشه ۷۴۰۴)، مداری ساده روی بردبرد تعریف کنیم که ورودی را ۶ بار معکوس میکند، پیش بینی میشود مقدار خروجی و ورودی برابر باشد چون زوج بار عمل معکوس شدن انجام میشود؛ اگرچه نرم افزار مذکور توانایی شبیه سازی ندارد.

۱-۳ شرح دستگاه‌ها و وسایل مورد استفاده

یک دستگاه کامپیوتر که روی آن نرم‌افزار Fritzing نصب شده باشد کافی است .

۴-۱ روش انجام آزمایش

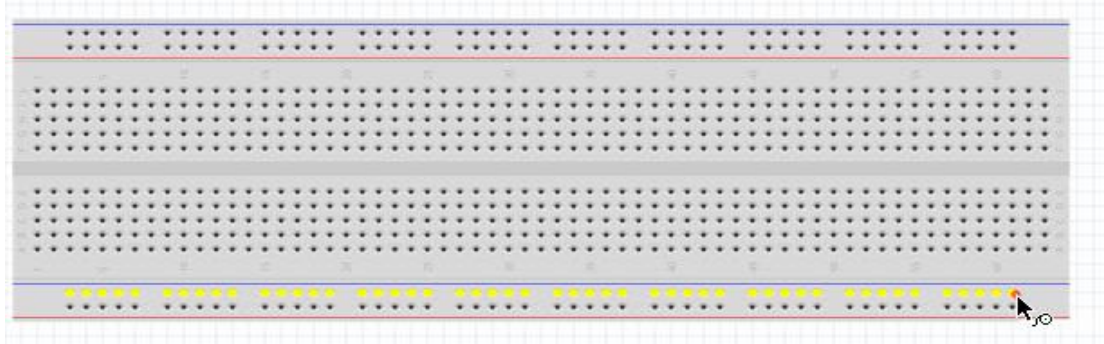
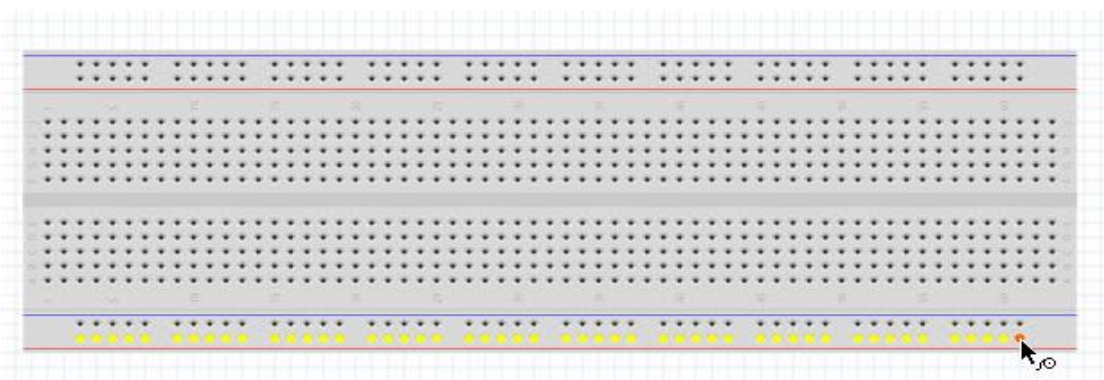
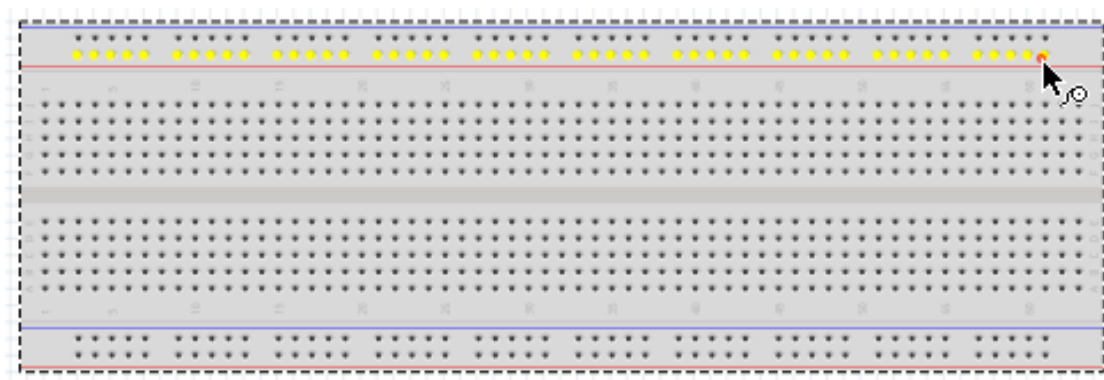
۴-۱-۱ بخش اول

برد مورد را با استفاده از قسمت سرچ نرم افزار پیدا میکنیم و میکشیم تا روی صفحه بیاید .

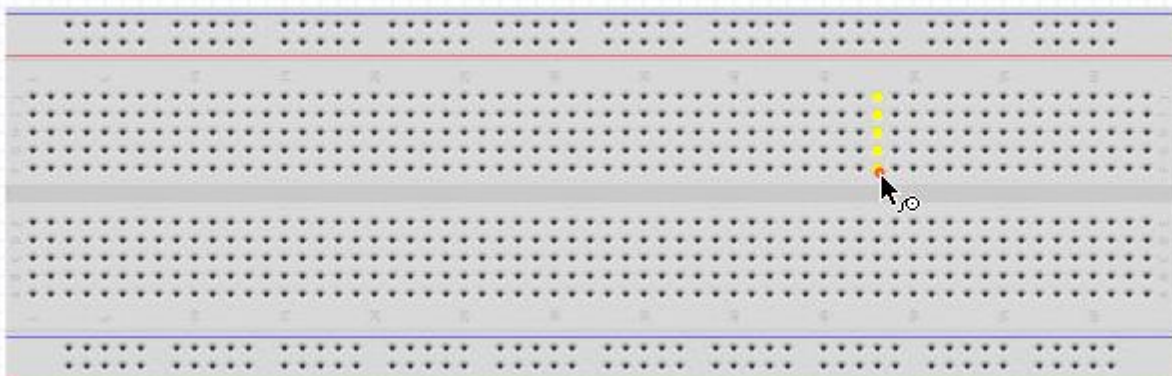
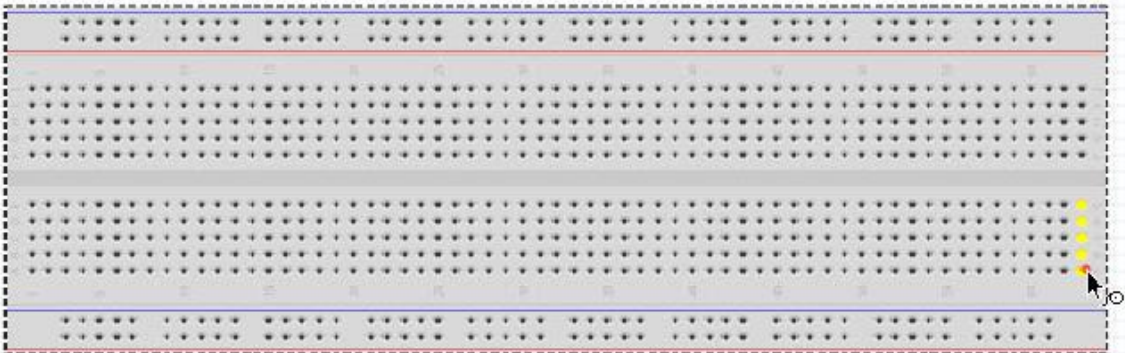
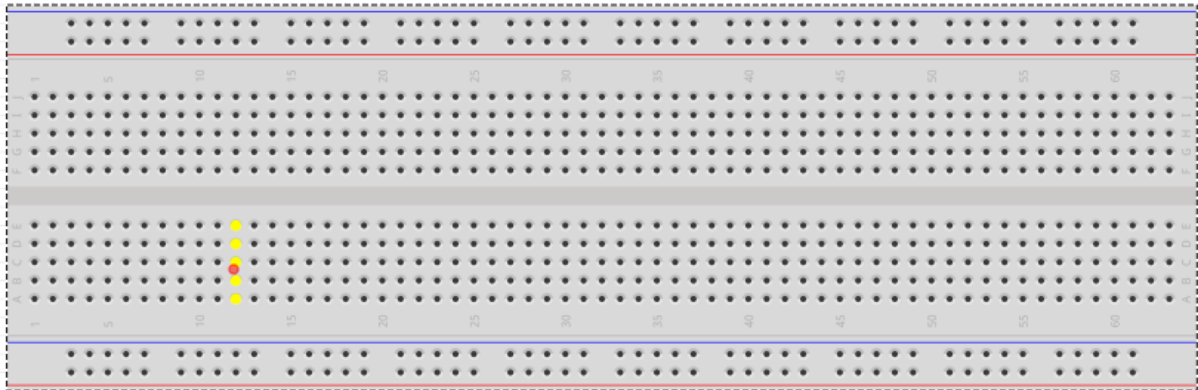
مانند شکل ، هر بردبرد دارای ۴ بخش از بالا به پایین است.

در نرم افزار فریتزینگ با نگه داشتن و کلیک چپ موس روی هر نقطه آن قرمز شده و تمامی نقاط متصل به آن زرد میشوند از این استفاده میکنیم و بر روی نقاط مختلف آزمایش میکنیم ، با توجه به نتایج آزمایش‌های انجام شده ، در بالاترین و پایین ترین بخش ها تمامی قسمت ها به صورت افقی به هم متصلند و میتوانیم از آنها برای اتصالات خود کمک بگیریم.

ابتدا بالاترین قسمت و پایین ترین قسمت را آزمایش میکنیم ، موس را بر روی نقطه ای خاص نگه داشته تا اتصالات را ببینیم.

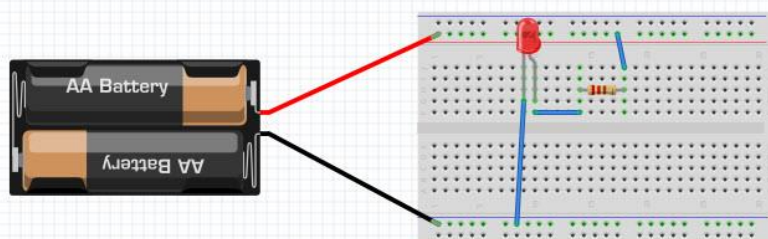


همان آزمایش را برای هر یک از بخش‌های میانی تکرار کرده و کلیک چپ موس را بر روی نقطه‌ای دلخواه در هر یک نگه میداریم. مشاهده میشود که قسمت‌های عمودی در همان نیمه به رنگ زرد درخواهند آمد. این بدین معنی است که در هر یک از بخش‌های میانی، نقاط به صورت عمودی به هم متصلند.



۱-۴-۲ بخش دوم

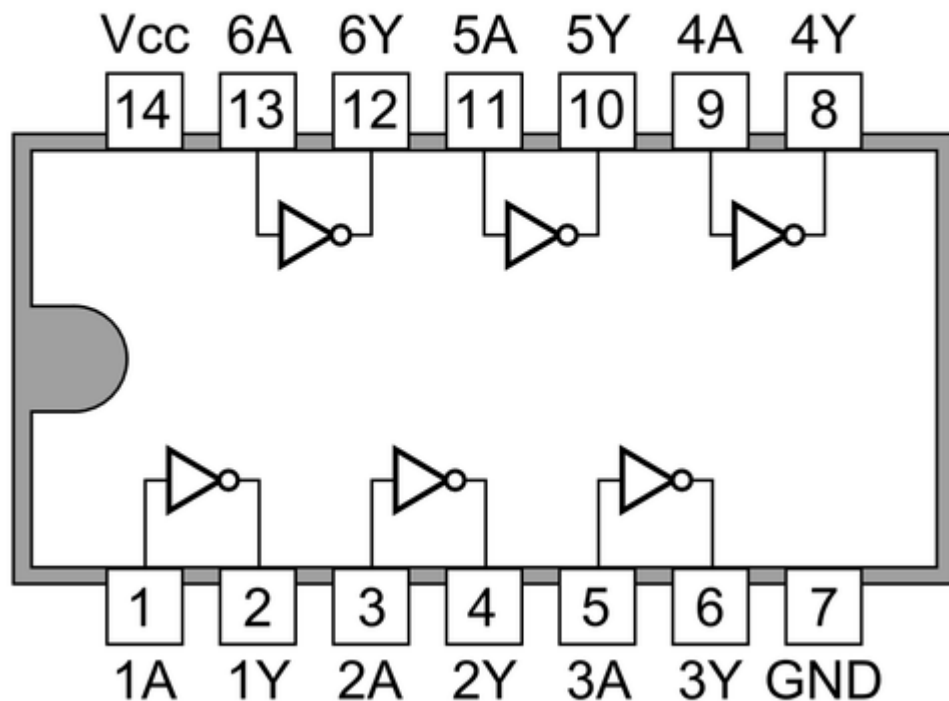
یک باتری، یک مقاومت و یک LED را به کمک قسمت سرچ نرم افزار پیدا کرده و میکشیم تا روی صفحه بیاید . در ابتدا دو سر باتری را به دو قسمت از بخش های بالایی و پایینی بردبرد متصل میکنیم تا یکی از آنها VCC و دیگری گرانند شود نیاز داریم بدانیم که کدام گرانند و کدام وی سی سی است زیرا در ادامه باید پایه کوتاه ال ای دی را به گرانند متصل کنیم ، مثبت یا منفی بودن قطب های باتری مشخص شده است اما اگر هم مشخص نشده بود بر روی شکل آن ، با نگه داشتن موس روی دو سر باتری میتوانستیم متوجه منفی یا مثبت بودن آن شویم. LED و پس از آن مقاومت را روی برد برد قرار میدهیم ، سر بلند LED را به یک سر مقاومت و سر دیگر مقاومت را به سی سی متصل میکنیم و سر کوتاه LED را نیز همان طور که گفته شده بود به گرانند متصل مینماییم و بدین ترتیب یک مدار ساده شامل مقاومت، باتری، LED و بردبرد تشکیل میدهیم.



۱-۴-۳ بخش سوم

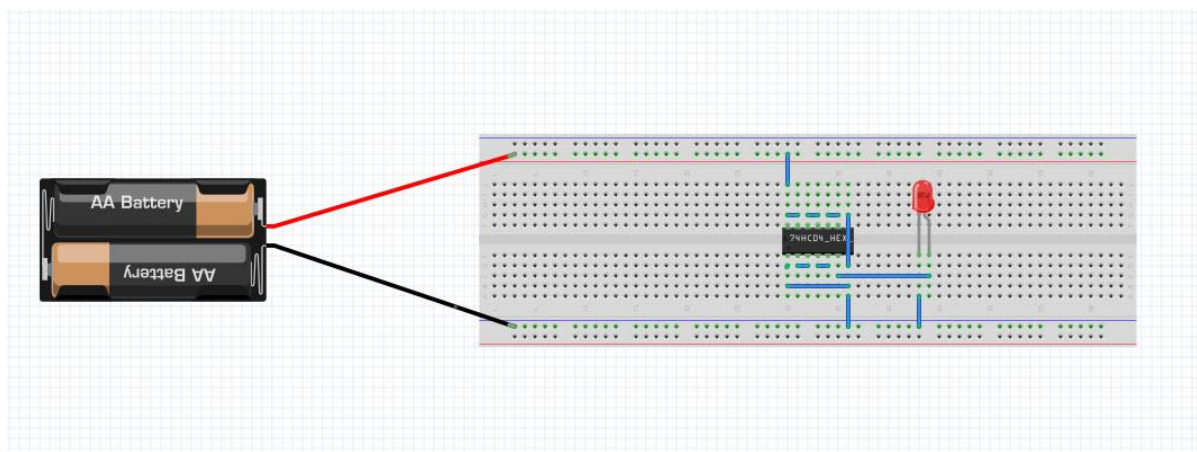
مطابق تصویر ابتدا یک تراشه ۷۴۰۴ که شامل ۶ عدد گیت نات یا معکوس کننده میباشد را از قسمت سرچ انتخاب کرده می‌آوریم آن را بین دو نیمه بخش میانی بردبرد قرار می‌دهیم. آن را نباید در یک نیمه قرار دهیم چون اینگونه تمام خروجی‌ها و ورودی‌های بالا و پایین به هم متصل میشوند و تراشه کاربرد خود را از دست میدهد. درباره تراشه توضیحات زیر را داریم.

7404 Hex Inverters



مطابق شکل پایه‌های ۱۴ و ۷ به وی سی سی و گراند باید متصل شوند و باقی پایه‌ها برخی ورودی و برخی خروجی، شش گیت نات هستند.

سپس یک LED انتخاب کرده رو برد برد قرار می‌دهیم. مانند قسمت قبل دو سر یک باتری را به بخش‌های بالایی و پایینی برد برد متصل می‌کنیم تا در ادامه از وی سی سی و گراند استفاده کنیم. در نهایت به کمک سیم‌های کوچک خروجی‌ها و ورودی‌ها را طوری به هم متصل می‌کنیم که ورودی ابتدایی از ۶ گیت نات عبور کرده و خروجی حاصل شده وارد یک LED شود که سر دیگر آن به قطب منفی باتری است.

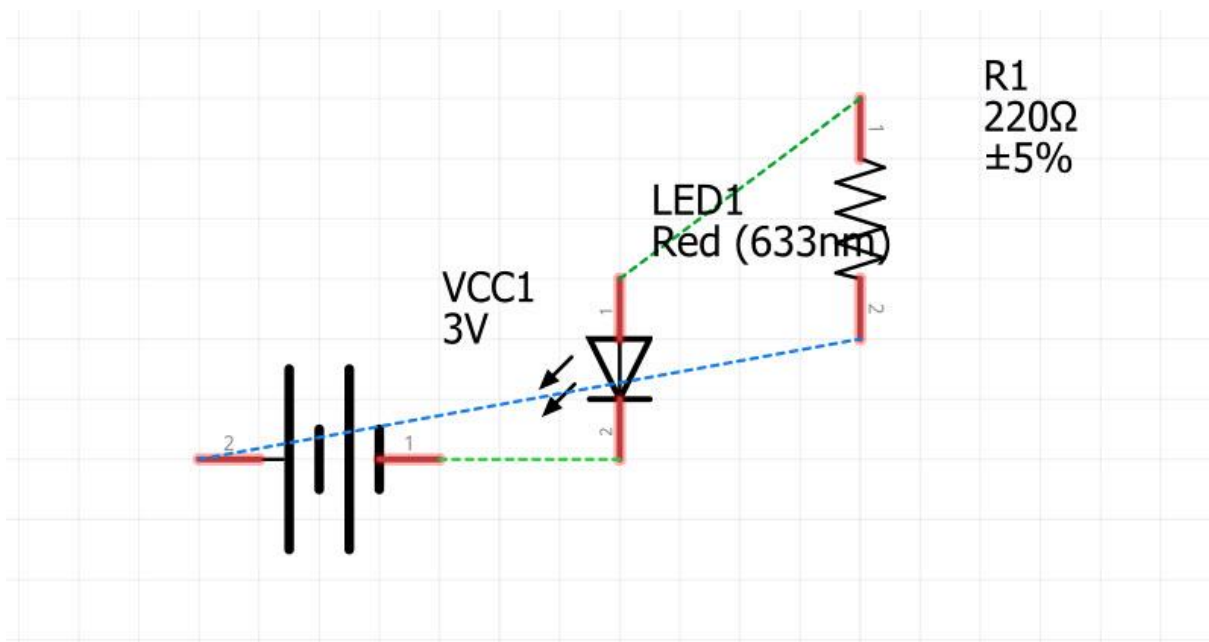


۱-۵ بحث و تفسیر اطلاعات و نتایج به دست آمده از آزمایش

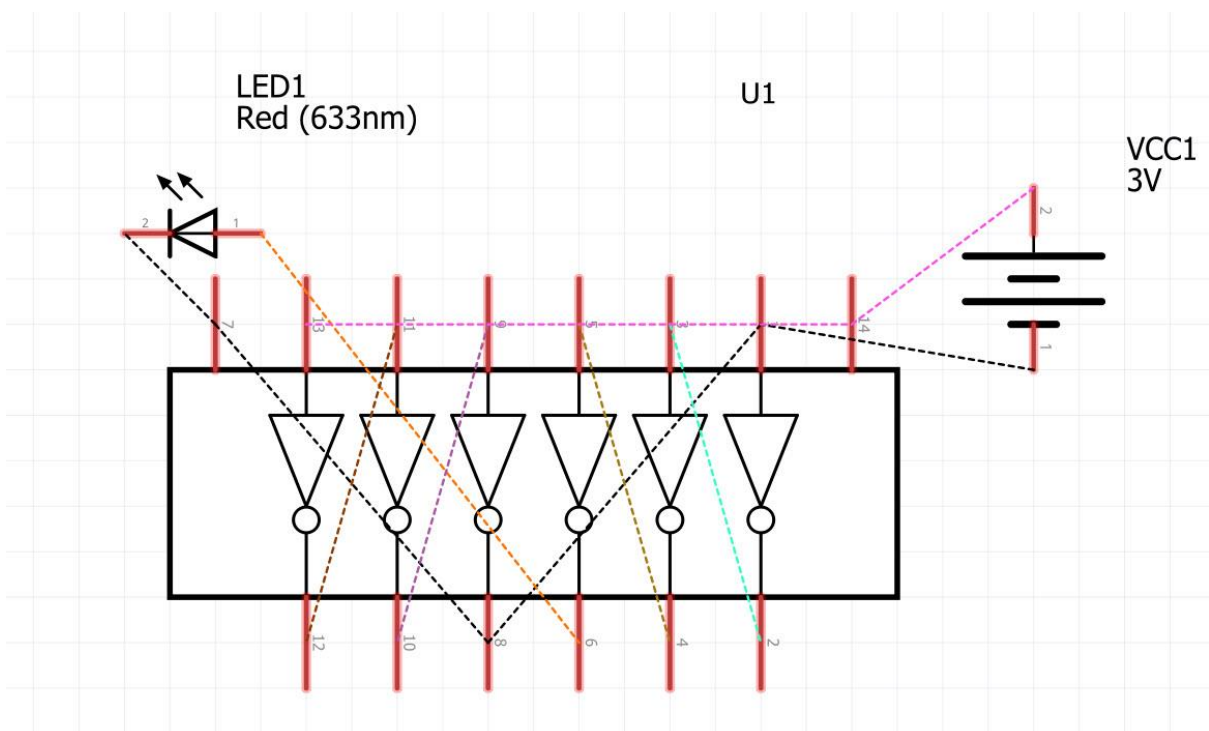
در هر یک از بخش‌ها با قرار دادن صفحه بر روی حالت شماتیک شکل شماتیک مدارها را میبینیم.

در بخش ۱ با توجه به تصاویر متوجه شدیم که اتصالات بالاترین و پایین‌ترین بخش‌های بردبرد به صورت افقی و اتصالات بخش‌های میانی به صورت عمودی میباشد. از طرفی متوجه شدیم که میتوان از بالاترین و پایین‌ترین بخش‌ها برای انتشار ولتاژهای مختلف (گراند و وی سی سی) استفاده کرد و بر روی بردبرد مدار بست.

در بخش ۲ همانطور که در شکل شماتیک به دست آمده میبینیم مدار موردنظر همانند خواسته سوال طراحی شده قطب منفی باتری به درستی به پایه منفی LED متصل شده و از یک مقاومت برای کنترل ولتاژ دو سر LED استفاده شده است.



در بخش ۳ با بررسی اتصالات متوجه میشویم که ورودی پس از ۶ بار نات شدن، به خروجی تبدیل شده و درواقع مقدار خروجی و ورودی یکسان خواهد بود و در شکل رسم شده LED روشن خواهد شد.



۶-۱ منابع و مراجع

- وبسایت ویکی‌پدیا

۲ آزمایش دوم: ساخت مدار با Logisim

۱-۲ مقدمه و هدف:

در این آزمایش در بخش اول می‌خواهیم یک full adder تک بیتی که رقم نقلی را هم محاسبه میکند را در نرم‌افزار Logisim شبیه سازی کنیم و در بخش دوم نیز می‌خواهیم یک جمع کننده / تفریق کننده ۴ بیتی را شبیه سازی کنیم.

هدف اصلی آزمایش‌های این بخش نیز آشنایی بیشتر با کارکرد و محیط نرم‌افزار Logisim است؛ نرم‌افزار Logisim یک نرم‌افزار برای شبیه سازی مدارها و نتایج زنده آنها میباشد.

۲-۲ تجزیه و تحلیل تئوری آزمایش‌ها

از درس مدار منطقی میدانیم که هر یک از C_{out} , sum چه فرمول هایی دارند :

$$sum = C_{in} \text{ XOR } A \text{ XOR } B \qquad C_{out} = ((B \text{ XOR } A) \text{ AND } C_{in}) \text{ OR } (A \text{ AND } B)$$

با توجه به فرمول‌های بالا جمع کننده کامل یک بیتی را پیاده سازی میکنیم.

در بخش دوم میدانیم برای ساخت یک جمع کننده/تفریق کننده که با یک بیت C_{in} نوع آن مشخص میشود میتوانیم C_{in} را با تمامی بیت های عدد دوم XOR کرده تا اگر C_{in} برابر ۱ بود نات شوند و با ۱ هم جمع شده و مکمل ۲ ایجاد شود و عمل تفریق به صورت جمع با مکمل ۲ صورت گیرد و نیز میدانیم که برای ساخت جمع کننده ۴ بیتی نیاز به ۴ جمع کننده ۱ بیتی داریم که کری اوت هر کدام به کری این بعدی وصل میشود غیر از کری این اولین فول ادر و کری اوت آخرین فول ادر که آزاد هستند.

۳-۲ شرح دستگاه‌ها و وسایل مورد نیاز

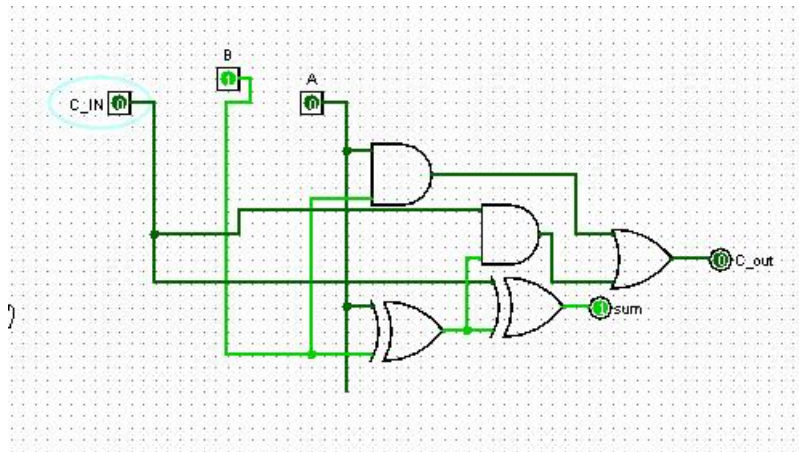
یک دستگاه کامپیوتر که روی آن نرم‌افزار Logisim نصب شده باشد.

۴-۲ روش انجام آزمایش

بخش اول :

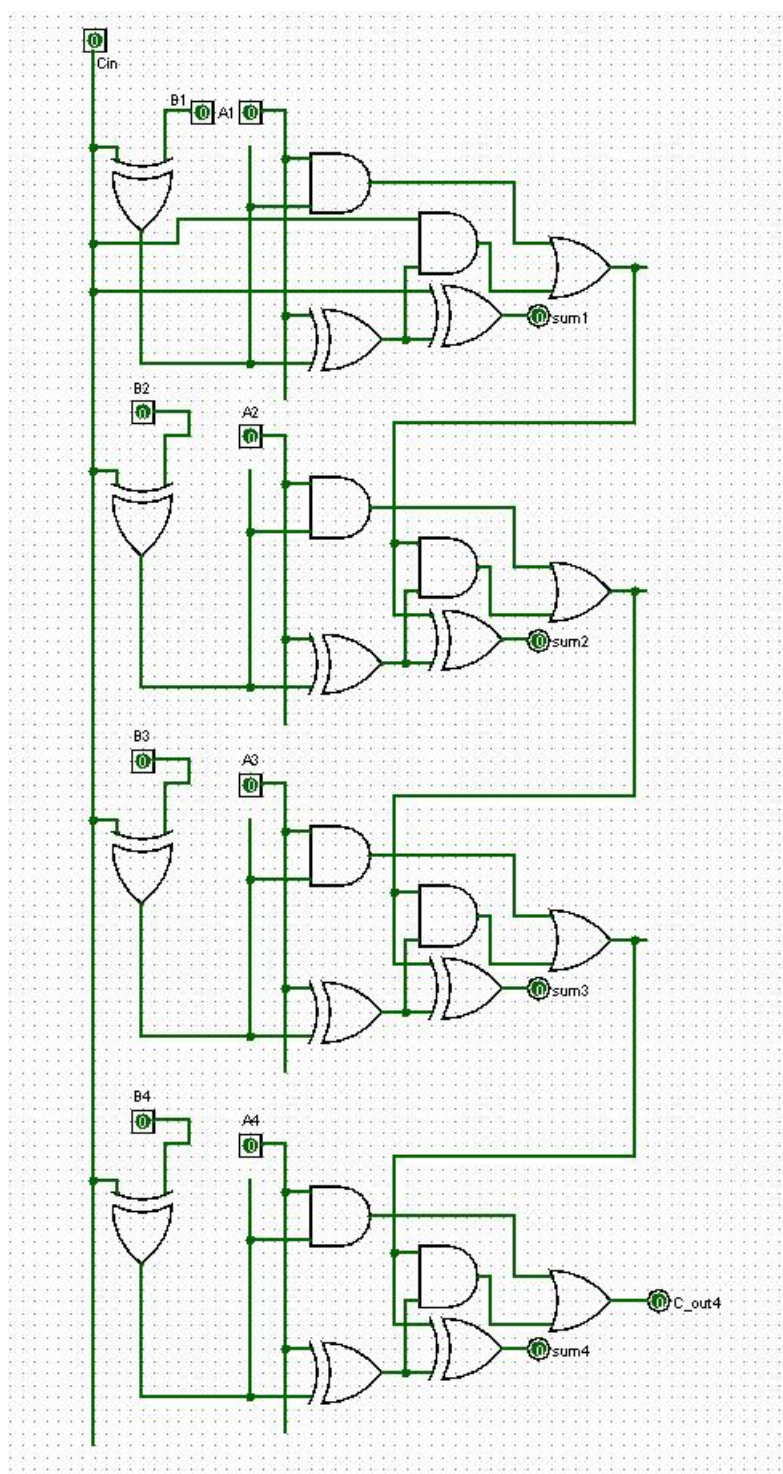
ابتدا ۳ پین به نامهای A , B , C_{in} به عنوان ورودی‌های جمع‌کننده ساخته تا بتوانیم به راحتی و با کلیک موس مقدار آنها را بین ۰ و ۱ تغییر دهیم. حال گیت‌های اند، اور و ایکس اور را به مدار اضافه کرده و در تنظیمات هر کدام تعداد ورودی‌هایشان را روی ۲ تنظیم می‌کنیم و سیم اضافه می‌کنیم تا مطابق فرمول‌ها خروجی‌ها را بسازیم.

در انتها دو عدد پراب به نامهای sum و $cout$ ساخته تا خروجی‌ها را تشکیل دهند.



در بخش دوم، به کمک ۴ مدار طراحی شده مانند بخش اول طبق توضیحات ارائه شده در بخش تئوری یک جمع‌کننده کامل ۴ بیتی ساخته و سپس به کمک بیت C_{in} و xor کردن آن با تک تک بیت‌های عدد دوم جمع‌کننده را به جمع‌کننده/تفریق‌کننده تبدیل می‌کنیم.

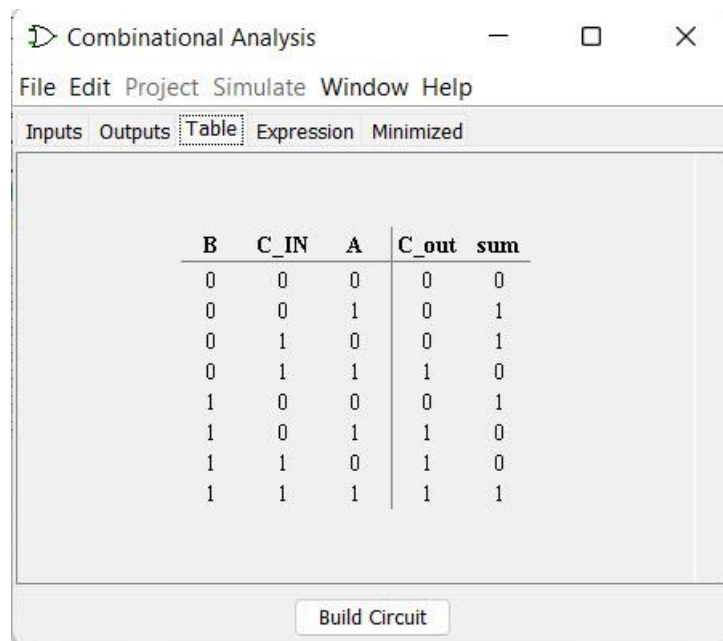
این عمل سبب می‌شود که اگر C_{in} برابر ۰ بود همان عملکرد عادی جمع کردن را داشته باشیم و اگر C_{in} برابر ۱ بود؛ مکمل دوی عدد دوم محاسبه شود و جمع شود که سبب حاصل تفریق محاسبه شود.



۵-۲ بحث و تفسیر اطلاعات نتایج به دست آمده از آزمایش

همانطور که مشاهده شد مطابق فرمول‌های بخش تئوری دو مدار جمع کننده یک بیتی کامل و جمع کننده/تفریق کننده ۴ بیتی در سطح گیت به کمک نرم‌افزار مربوطه شبیه سازی شده و به راحتی میتوان با چند کلیک خروجی های مختلف به ازای ورودی های مختلف را در مدارها چک کرد. در نهایت برای مطمئن شدن از کارکرد درست مدارها به بررسی جدول صحت تولید شده توسط نرم‌افزار که از بخش

project -> analyze circuit->tables قابل دسترسی است، بررسی میکنیم.



B	C_IN	A	C_out	sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Cin	B1	A1	B2	A2	B3	A3	B4	A4	sum1	sum2	sum3	C_out4	sum4
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	0	1
0	0	0	0	0	0	0	1	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	0	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	0	1	1	0
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	0	1	0	1
0	0	0	0	0	1	0	1	0	0	0	1	0	1
0	0	0	0	0	1	1	0	0	0	0	0	1	0
0	0	0	0	0	1	1	0	1	0	0	0	1	0
0	0	0	0	0	1	1	1	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	1	0	1	0	0	1
0	0	0	0	1	0	0	1	0	0	1	0	0	1
0	0	0	0	1	0	0	1	1	0	1	0	1	0
0	0	0	0	1	0	1	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0	1	0	1	1	0	1
0	0	0	0	1	1	0	1	0	0	1	1	1	0
0	0	0	0	1	1	1	0	0	0	1	0	0	1
0	0	0	0	1	1	1	1	0	0	1	0	1	0
0	0	0	0	1	1	1	1	1	0	1	0	1	1
0	0	0	1	0	0	0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0	1	0	1	0	0	1
0	0	0	1	0	0	0	1	1	0	1	0	1	0
0	0	0	1	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	0	1	1	0	1	1	0	1
0	0	0	1	0	0	1	1	1	0	1	1	1	0
0	0	0	1	0	1	0	0	0	0	1	1	0	0
0	0	0	1	0	1	0	0	1	0	1	1	0	1
0	0	0	1	0	1	0	1	1	0	1	1	1	0
0	0	0	1	0	1	1	0	0	0	1	0	0	1
0	0	0	1	0	1	1	0	1	0	1	0	1	0

۲-۶ منابع و مراجع:

- وبسایت ویکی‌پدیا

۳ آزمایش سوم: ساخت مدار با Proteus

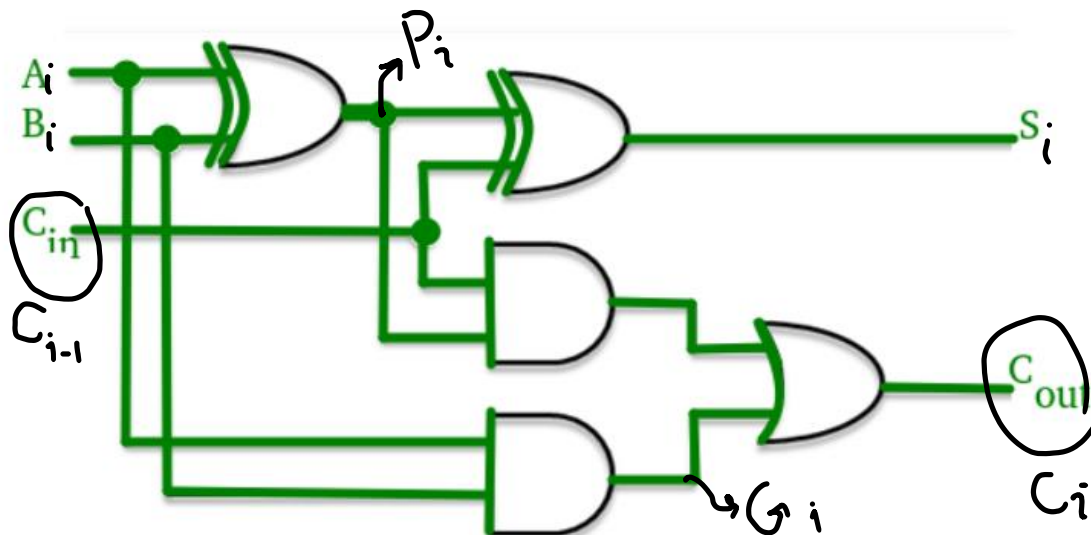
۳-۱ مقدمه و هدف

در این آزمایش قصد داریم یک جمع کننده carry-look ahead ۴ بیتی را در سطح گیت با استفاده از نرم‌افزار Proteus طراحی کنیم.

نرم‌افزار Proteus یک نرم‌افزار قدرتمند برای شبیه‌سازی مدارهای مختلف و بررسی و تحلیل آنها می‌باشد. هدف اصلی از این آزمایش آشنایی ابتدایی با محیط نرم‌افزار Proteus و کارکرد آن می‌باشد.

۳-۲ تجزیه و تحلیل تئوری آزمایش

میدانیم که برای طراحی جمع‌کننده مورد نظر در سطح گیت باید از فرمول‌های زیر استفاده کنیم که در آنها i از ۰ تا ۳ می‌باشد (زیرا جمع‌کننده ۴ بیتی است). در قسمت قبل یک فول ادر را طراحی کردیم و گفتیم که به چه صورت است



به قسم زیر پی‌آی و جی‌آی را تعریف می‌کنیم

$$P_i = A_i \text{ xor } B_i$$

$$G_i = A_i \text{ and } B_i$$

سپس داریم :

$$S_i = P_i \text{ xor } C_{i-1}$$

$$C_i = G_i \text{ OR } (P_i \text{ AND } C_{i-1})$$

رقم C4 رقم نقلی یا همان cout مدار و رقم C0 کری ورودی مدار خواهد بود.

اگر تلاش کنیم همه‌ی سی‌آی‌ها را به گونه‌ای ریزایت کنیم که تنها سی موجود در آنها سی صفر باشد، سپس می‌توانیم اس‌آی‌ها را به شکل ایکس‌اور پی‌آی و سی‌آی منهای یک پیاده‌سازی کنیم. خوبی این

مدار این است که سی‌ها خیلی زود آماده شده و تا خیر مدار بسیار کم میشود :

$$\begin{aligned}C_1 &= G_0 + P_0 C_{in} \\C_2 &= G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_{in} \\C_3 &= G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{in} \\C_4 &= G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{in}\end{aligned}$$

حال در ادامه بر اساس این توضیحات مدار را میسازیم.

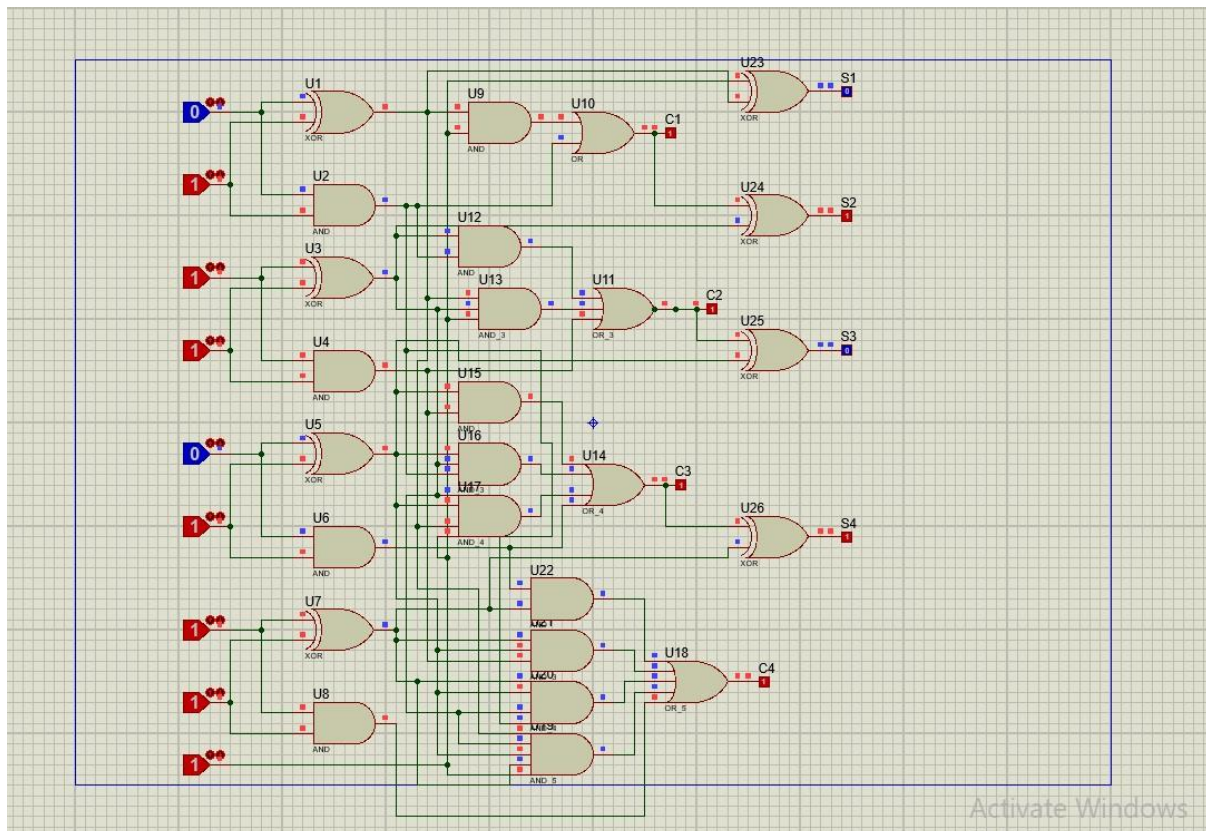
۳-۳ شرح دستگاه‌ها و وسایل مورد نیاز

یک دستگاه کامپیوتر که روی آن نرم‌افزار Proteus نصب شده میباشد.

۳-۴ روش انجام آزمایش

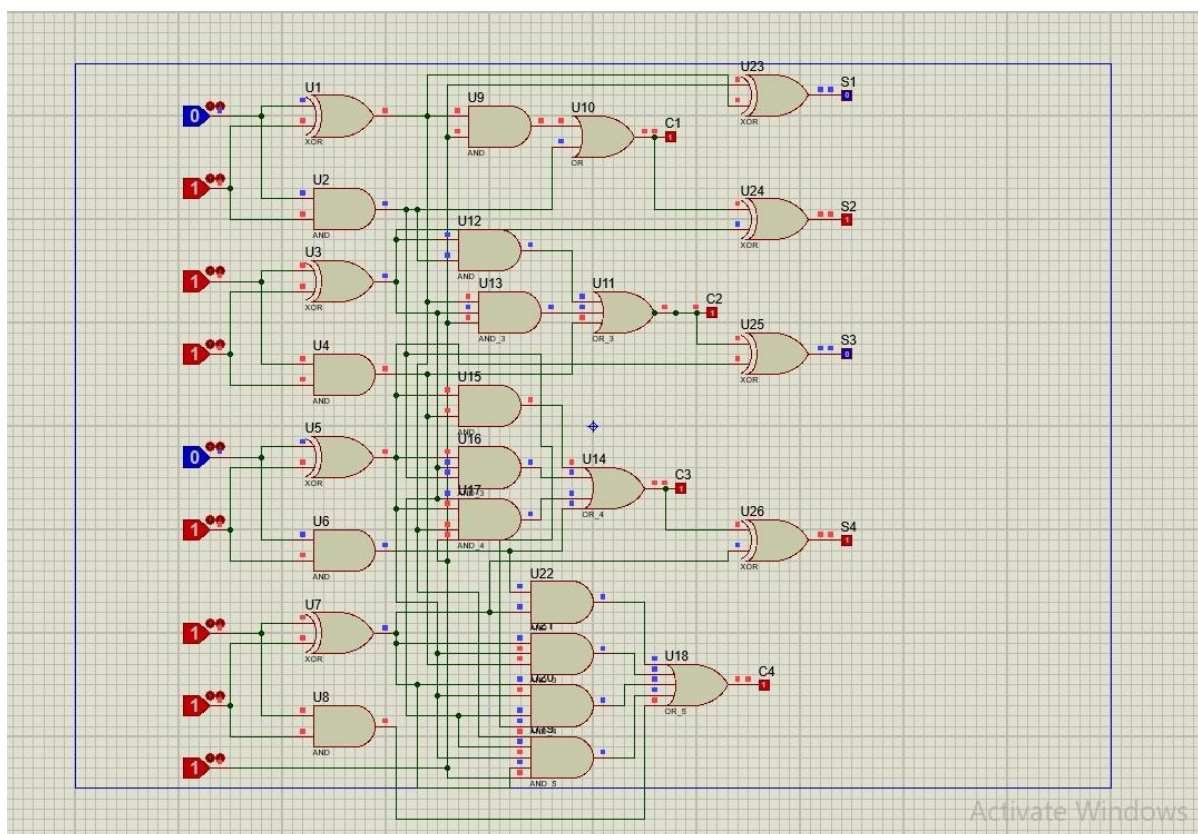
برای اضافه کردن اجزای مختلف مدار در بخش کامپوننت مود روی P کلیک میکنیم و سپس اسم گیت، تراشه و ... مورد نظر را سرچ کنیم ، سپس آن اسم در ویندیوی زیر پی ذخیره شده و بار بعدی از اونجا برش میداریم.

برای ورودی‌های از logicstate استفاده میکنیم که با یک کلیک به راحتی میتوان مقدار آنها را بین ۰ و ۱ عوض کرد و برای خروجی‌ها از logicprob استفاده میکنیم که خروجی را به راحتی به صورت ۰ و ۱ به ما نشان میدهد. در نهایت به کمک گیت‌های اند، اور و ایکس‌اور طبق فرمول مذکور در بخش تئوری مدار را طراحی میکنیم.



۳-۵ بحث و تفسیر نتایج به دست آمده از آزمایش

با کمک فرمولهای بخش تئوری یک جمع‌کننده ۴ بیتی از نوع کری لوک اهد شبیه سازی کردیم اکنون اگر به ورودی‌ها مقدار دهیم و دکمه استارت را بزنیم میتوانیم نتیجه شبیه سازی مدار را ببینیم با انجام اینکار به ازای ورودی‌های مختلف خروجی‌های نظیر و مورد نظر را به دست میاوریم مانند مثال زیر که معادل جمع دو عدد ۱۰۱۰ و ۱۱۱۱ با کری ورودی ۱ میباشد.



۳-۶ منابع و مراجع

- وبسایت گیکز فور گیکز
- وبسایت یوتیوب
- وبسایت ویکی‌پدیا